#29

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/259,991	Confirmation No. 5948
Appl. No. Bar Code	:		
Applicant	:	Mahne	
Filed	:	03-01-1999	
TC/A.U.	:	2134	
Examiner	:	Matthew Smithers	
Docket No.	:	M000-P02003US	
Customer No.	:	33356	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The following Appeal Brief is submitted pursuant to the Notice of Appeal dated August 5, 2003. The following Appeal Brief is submitted in triplicate pursuant to 35 C.F.R. § 1.192 for consideration by the Board of Appeals and Interferences.

I. REAL PARTY IN INTEREST

The real party in interest is MAZ Technologies, Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no Appeals or Interferences which will affect or be affected by the outcome of this Appeal.

Re-examination number 90/006,529, ordered February 10, 2003, may directly affect, be directly affected by, or have a bearing on the Board's decision.

III. STATUS OF THE CLAIMS

Claims 59, 68-73, 75, 76, 78, and 79 were rejected and are pending. Claims 47-58, 60-67, 74, 77, and 80 were allowed.

IV. STATUS OF THE AMENDMENTS

An amendment to the claims dated July 7, 2003 was filed subsequent to the final rejection dated May 5, 2003. An amendment to claim 63 incorporated all the limitations of base claims 60-62 to put it in independent form and to overcome an objection to being dependent upon a rejected base claim. The amendment was entered. Claim 80 was a new claim, dependent from claim58, including an additional limitation. The amendment was entered.

V. SUMMARY OF THE INVENTION

A method of decrypting an electronic file that is to be opened in an application program running in a suitable environment for operating the program (p. 8, lines 6-12) comprising the steps of: issuing an open document command to act upon the file (p. 11, lines 19-20); intercepting the open document command (p. 12, lines 1-2); retrieving a decryption key value (p.

12, lines 15-16); selecting an algorithm to use with the file from one of a plurality of algorithms (p. 13, lines 12-14); inputting a decryption key with a key value (p. 9, lines 3-10, p. 12, lines 18-22); validating the decryption key value with the key value associated with a file identifier (p. 14, lines 3-13); using the key value and the selected algorithm to decrypt the file to an unencrypted file (p. 14, lines 13-15); running a virus scan program on the decrypted file (p. 12, lines 6-8); completing the open document command by performing the open document command upon the unencrypted file instead of the file (p. 13, lines 4-6).

A method of encrypting and decrypting a file with one of a plurality of algorithms (p. 8, lines 6-12, p. 13, lines 12-14) comprising the steps of: selecting an algorithm to use with the file from the plurality of algorithms (p. 13, line 16 – p. 14, line 2); selecting an encryption key with a key value (p. 11, lines 4-7); generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file (p. 14, lines 3-5); adding the file identifier to the file (p. 14, lines 5-7); inputting a decryption key with a decryption key value (p. 9, lines 3-10, p. 12, lines 18-22); validating the decryption key value with the key value associated with the file identifier (p. 14, lines 3-13); and using the key value and the selected algorithm to decrypt the file (p. 14, lines 13-15), wherein the file is located in a document or image repository (p. 15, lines 14-16).

A method of encrypting and decrypting a file with one of a plurality of algorithms (p. 8, lines 6-12, p. 13, lines 12-14) comprising the steps of: selecting an algorithm to use with the file from the plurality of algorithms (p. 13, line 16 – p. 14, line 2); selecting an encryption key with a key value (p. 11, lines 4-7); generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file (p. 14, lines 3-5); adding the file identifier to the file (p. 14, lines 5-7), wherein a portion of the file

identifier is encrypted before it is inserted into the file (p. 14, lines 20-24); running a virus scan program on the file before it is encrypted (p. 10, lines 8-10); using the key value and the selected algorithm to encrypt the file and generate an encrypted file(p. 11, lines 15-17, p. 13, lines 12-14); inputting a decryption key with a decryption key value (p. 9, lines 3-10, p. 12, lines 18-22); validating the decryption key value with the key value associated with the file identifier (p. 14, lines 3-13); using the key value and the algorithm to decrypt the file (p. 14, lines 13-15).

A method of encrypting and decrypting a file with one of a plurality of algorithms (p. 8, lines 6-12, p. 13, lines 12-14) comprising the steps of: receiving an encrypted file from a first person by a second person in an e-mail message (p. 9, lines 18-22, p. 15, lines 16-19); extracting a file identifier from the file (p. 14, lines 3-11); inputting a decryption key with a decryption key value (p. 9, lines 3-10, p. 12, lines 18-22); validating the decryption key value with a key value associated with the file identifier (p. 14, lines 3-13); using the key value and the selected algorithm to decrypt the file (p. 14, lines 13-15).

VI. ISSUES PRESENTED

The following issues are presented by this Appeal:

Are claims 68-73, 76, 78 and 79 unpatentable under 35 U.S.C. § 103(a) as rendered obvious by Hsu (USP 5,584,023) in view of Brundrett et al. (USP 6,249,866)?

Are claims 59 and 75 unpatentable under 35 U.S.C. § 103(a) as rendered obvious by Hsu (USP 5,584,023) in view of Brundrett et al. (USP 6,249,866) and Finley (USP 5,815,571)?

VII. GROUPING OF THE CLAIMS

Applicants submit that the claims do not stand or fall together. Accordingly, the claims are to be grouped as follows:

Group A	Claims 68-73, 79
Group B	Claim 76 and 78
Group C	Claim 59
Group D	Claim 75

VIII. ARGUMENT

A. Preface

It is respectfully requested that the Board of Patent Appeals and Interferences consider the inadequacy of the quality of this examination. For a claim rejection under 35 U.S.C § 103(a) to be proper, the examiner has the burden of establishing *a prima facie case of obviousness*.

The Examiner did not and can not meet the burden of establishing *a prima facie case of obviousness*. The Examiner relied on primary reference Hsu (USP 5,584,023). The Examiner merely recited the overall utility of Hsu and did not describe how the steps taught in Hsu relate to the specific steps of the claimed method. Further, when the Examiner set forth rejections based on Hsu in light of Brundrett (USP 6,249,866) and Finley (USP 5,815,571), there was no explanation of how the individual features of Hsu, Brundrett and Finley can either be combined

or modified to arrive at the entirety of the steps of the claimed methods. The Examiner is correct that one or two features such as the utility of encryption and the utility of a virus scan may found in Hsu, Brundrett and Finley. However, the mere fact that a generic utility is common does not render the claimed methods obvious.

Following each Office action rejection, a detailed response was provided showing why each and every rejection was improper. The Examiner did allow some claims. However, following each response, the Examiner issued another Office action, substantially changing the basis for rejection, but never establishing *a prima facie case of obviousness*. At best, the Examiner was able to show that the prior art disclose one of the plurality of steps of the claimed methods. However that step was merely a description of the claimed inventions' generic utility.

B. Overview of Hsu

Hsu is directed to a computer system including a transparent and secure file transfer mechanism. In Hsu's computer system, all encryption and decryption is performed using a single algorithm.

C. Overview of Brundrett

Brundrett is directed to a file system which includes transparent file encryption and decryption capabilities. Brundrett teaches that the user can choose among available encryption algorithms.

D. Overview of Finley

Finley is directed to a computer system with secure data paths and a method of protection. Finley's object is to prevent harm from viruses, essentially by quarantining all incoming data

before allowing the data to be moved to a normal workspace. Finley teaches that, instead of a single computer, there should be three computers: a main computer, a security computer and a test computer. Finley refers to the security computer as a "firewall." All security functions are implemented in the security computer, which "must not" execute user programs. Before a new program which has been downloaded from the Internet can be run on the main computer, it is first run on the test computer.

E. **Group A: Rejection of Claims 68-73 and 79 as Unpatentable over Hsu in view of Brundrett**

The Examiner asserted that all the limitations of the claims in this group are taught by the combination of Hsu and Brundrett. Yet, the combination of these references fails to teach or suggest all of the limitations recited in the claims in this group.

The invention of claims 68-73 and 79 recite among other limitations:

> inputting a decryption key with a decryption key value;
>
> validating the decryption key value with the key value associated with the file identifier.

In fact, these steps are not disclosed, taught or suggested by Hsu or Brundrett. The Examiner asserted:

> Hsu does teach or suggest the above along with teaching the through the transformation, data values (data identifiers) are created and are later used to generate the decryption index values (decryption key) needed in the validation process (see column 14, lines 50-67).

This does not support the rejection. Hsu's validation process compares a text string that resides both in the enode and the in-core inode [See Column 14, lines 50-54]. Hsu's encryption method requires that the user enter a password key. This password key and a random mapping seed table are processed through a shuffle function that generates both the encryption table and the decryption table [See Column 11, line 45 – Column 12, line 29]. The password key is encrypted using the encryption table and is then appended to both the file's enode and the in-core inode [See Column 12, line 50-53, Column 14, lines 42 – Column 15, line 4 and Column 17, lines 4-13].

Hsu's validation process includes decrypting the enode using the decryption key and then comparing the text string located in the "magic_text" field of the decrypted enode to the corresponding text string located in the in-core inode [See Column 13, lines 56-65 and Column 14, lines 53 – Column 15 line 4]. Hsu's validation compares the data in the "magic_text" field which is the user entered password key. The user entered password key is not the encryption table (encryption key) and is not the decryption table (decryption key).

Unlike claims 68, 69, 71 and 79, Hsu's validation process does not utilize the decryption key as an element that is compared to another element. Therefore Hsu failes to teach or suggest the limitations recited in claims 68, 69, 71 and 79.

The Examiner cited nothing in Brundrett that either on its own or in combination with Hsu would teach the steps recited in the claims of this group. Therefore, the claims of this group are not obvious from Hsu in view of Brundrett.

For the reasons set forth in this section, the claims in this group are patentable over the combination of Hsu and Brundrett. As such, the obviousness rejection of the claims in this group should be overturned.

F. Group B: Rejection of Claims 76 and 78 as Unpatentable over Hsu in view of Brundrett

The Examiner asserts that all the limitations of the claims in this group are taught by the combination of Hsu and Brundrett. However, the combination of these references fails to teach or suggest all of the limitations recited in the claims in this group.

The invention of claims 76 and 78 recite among other limitations:

> generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file.

The Examiner asserted:

> Hsu does teach or suggest appending an enode data structure (file identifier) to a regular file after the file has been transformed through the use of an encryption table which serves as the encryption key. The encryption table (encryption key) is formed through a shuffling / index value substitution function applied to the password key and seed table (see column 12, line 25 to column 12, line 26). Through this process data values (data identifiers) are created and are associated arithmetically to the decryption index values of the decryption table (see column 12, lines 27-49). The

> contents of the identified enode structure (file identifier) can be used in the authentication of the encrypted data. This file identifier has associated data identifiers which are part of the encryption table (encryption key).

This does not support the rejection. Hsu's data values are not generated based on the file, but are created by the "shuffle function" based only on the user entered "password key" and a "predefined seed table" [See Column 11, line 45 – Column 12, line 49]. Therefore Hsu fails to teach or disclose the step of "generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file". Hence, Claims 60, 76, and 77 are not obvious from Hsu.

Yet, the Examiner asserted:

> When Brundrett verifies the integrity and source of the encryption key, the data structure which holds data identifiers, is compared to verify the structure (see column 16, lines 5-16).

This does not support the rejection. Brundrett is merely validating the session key encryption / decryption of the File Encryption Key which is passed between the EFS driver (Figure 46) and the EFS service (Figure 50) [See Column 4, line 54 – Column 5, line 9 and Column 15, line 51 – Column 16, line 16].

In fact, there are three primary encryption / decryption schemes within Brundrett. One encryption / decryption scheme uses the File Encryption Key (FEK) to encrypt / decrypt the user's file [See Column 9, line 66 – Column 10 line 1]. The second encryption scheme uses the Public Key of the user to encrypt the File Encryption Key and the Private Key of the user to

decrypt the FEK [See Column 10, lines 10-14]. The third encryption scheme is where the EFS driver (Figure 46) and EFS service (Figure 50) need to communicate. In the third scheme, a symmetric Session Key (SK) is utilized to encrypt data that is communicated between the EFS driver (Figure 46) and the EFS service (Figure 50) [See Column 4, Lines 54-61].

Within Brundrett, when a file is initially created and is designated to be saved in a pre-designated encrypted directory of a non-volatile disk, the FSCTL_SET_ENCRYPTION command is issued in order to turn on the encryption bit for a stream [See Column 16, lines 5-9]. The FSCTL_SET_ENCRYPTION command is accompanied by a data structure (Figure 8) containing:

> A public code so that NTFS 28 can differentiate between the two types of FSCTL calls;
>
> An EFS subcode to more particularly define the operation for the EFS driver 46 and/or the EFS service 50;
>
> EFS data containing the FEK;
>
> The FEK encrypted with the Session Key;
>
> Optionally EFS metadata.

[See Column 16, Lines 9-11 and Column 15, line 60 – Column 16, line 1].

Except for the public code, the information within the data structure is encrypted with the session key. [See Column 16, line 1 – Column 16, line 4]. Within the data structure, "The FEK encrypted with the Session Key" is also encrypted. In simple terms, the FEK is listed in the data

structure both in raw form and session key encrypted form. Then the data structure, as a whole, is encrypted with the session key.

In Brundrett, for purposes of verifying the integrity of the data structure (note that the EFS driver (Figure 46) and the EFS service (Figure 50) pass both the FSCTL_SET_ENCRYPTION command and the data structure, containing the FEK, between each other), the data structure is decrypted utilizing the symmetric system key. After the data structure has been decrypted, it contains both a FEK and a session encrypted FEK. The session encrypted FEK is then decrypted utilizing the symmetric system key. The verification step is accomplished by comparing the FEK values described in this paragraph. If they match, then the communication between the EFS driver (Figure 46) and the EFS service (Figure 50) was not compromised during the session key encryption and decryption process [See Column 16, line 12 – Column 16, line 16].

Nothing in Brundrett regarding the verification of the session key encryption and decryption of the File Encryption Key teaches or suggests a method including the step of "generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file." Even if a person skilled in the art of transparent encryption and decryption methods combined Hsu and Brundrett, nothing is disclosed in either and nothing is taught in either to "generat[e] a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file." Therefore, the claims of this group are not obvious from Hsu in view of Brundrett.

Moreover, combining the teaching of Hsu (comparing the decrypted password key that was encrypted in the enode with the decrypted password key that was encrypted in the in-core inode to validate accurate decryption of a file) with the teaching of Brundrett (validating the session key decryption of the File Encryption Key when the File Encryption Key has been passed among the EFS driver and the EFS service) is improper because *the combination or modification would destroy Brundrett's intended function.*

If Brundrett's validation focused on the user entered password key of Hsu, there would be no validating if the passing of the File Encryption Key between the EFS driver and the EFS service was compromised. Therefore it would be improper to combine Brundrett and Hsu.

For the reasons set forth in this section, the claims in this group are patentable over Hsu in light of Brundrett. As such, the obviousness rejection of the claims in this group should be overturned.

G. Group C: Rejection of Claim 59 as Unpatentable over Hsu in view of Brundrett and Finley

The Examiner asserts that all the limitations of the claims in this group are taught by the combination of Hsu, Brundrett and Finley. However, the combination of these references fails to teach or suggest all of the limitations recited in the claim in this group.

Claim 59 is allowable for the same reasons as claims 68, 69, 71 and 79. Claim 59 recites among other limitations:

> inputting a decryption key with a decryption key value;
>
> validating the decryption key value with the key value associated with the file identifier.

Since Finley does not teach or suggest the above listed steps and nothing in Finley, when used in conjunction with Hsu and Brundrett would teach or suggest the above listed steps, the claim of this group is not obvious from Hsu in view of Brundrett and Finley.

For the reasons set forth in this section, the claim in this group is patentable over the combination of Hsu, Brundrett and Finley. As such, the obviousness rejection of the claim in this group should be overturned.

H. Group D: Rejection of Claim 75 as Unpatentable over Hsu in view of Brundrett and Finley

The Examiner asserts that all the limitations of the claim in this group are taught by the combination of Hsu, Brundrett and Finley. However, the combination of these references fails to teach or suggest all of the limitations recited in the claim in this group.

Claim 75 is allowable for the same reasons as claim 71. Claim 75 contains each and every step and limitation of claim 71. Claim 75 recites among other limitations:

> running a virus scan program on the file before it is encrypted;
>
> validating a decryption key value with the key value associated with the file identifier;
>
> using the key value and the algorithm to decrypt the file.

Since Finley does not teach or suggest the above listed steps and nothing in Finley, when used in conjunction with Hsu and Brundrett would teach or suggest the above listed steps, the claim of this group is not obvious from Hsu in view of Brundrett and Finley.

For the reasons set forth in this section, the claim in this group is patentable over the combination of Hsu, Brundrett and Finley. As such, the obviousness rejection of the claim in this group should be overturned.

IX. INFORMATION DISCLOSURE STATEMENT

An information disclosure statement was submitted on July 2, 2003 with the fee set forth in 37 C.F.R. § 1.17(p). The Examiner properly identified that a statement required by 37 C.F.R. §1.97(e) was required in order to consider the information disclosure statement. The omission was in error and was not intentional. The Undersigned has made a *bona fide* attempt to comply with the rules. The Undersigned respectfully request that the information disclosure statement be considered.

X. CONCLUSION AND RELIEF

In view of the foregoing, it is believed that all claims patentably define the subject invention over the prior art of record and are in condition for allowance. The Undersigned request that the Board overturn the rejection of all claims and hold that all of the claims of the above referenced application are allowable.

Respectfully submitted,

SoCal IP Law Group

Date: November 12, 2003

Steven C. Sereboff, Reg. No. 37,035

Joel G. Landau, Reg. No. 54,732

SoCal IP Law Group
310 N. Westlake Blvd., Suite 120
Westlake Village, CA 91362
Telephone: 805/230-1350
Facsimile: 805/230-1355
email: info@socalip.com

APPENDIX

The claims involved in this Appeal are as follows:

59. A method of decrypting an electronic file that is to be opened in an application program running in a suitable environment for operating the program, comprising the steps of:

a) issuing an open document command to act upon the file;

b) intercepting the open document command;

c) retrieving a decryption key value;

d) decrypting the file using the decryption key value to create an unencrypted file; and

e) completing the open document command by performing the open document command upon the unencrypted file instead of the file; and

wherein steps c) and d) further comprise the steps of:

selecting an algorithm to use with the file from one of a plurality of algorithms;

inputting a decryption key with a key value;

validating the decryption key value with the key value associated with a file identifier;

using the key value and the selected algorithm to decrypt the file; and

running a virus scan program on the decrypted file.

68. A method of encrypting and decrypting a file with one of a plurality of algorithms, comprising the steps of:

selecting an algorithm to use with the file from the plurality of algorithms;

selecting an encryption key with a key value;

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file;

adding the file identifier to the file;

inputting a decryption key with a decryption key value;

validating the decryption key value with the key value associated with the file identifier; and

using the key value and the selected algorithm to decrypt the file;

wherein the file is located in a document or image repository.

69. A method of encrypting and decrypting a file with one of a plurality of algorithms, comprising the steps of:

selecting an algorithm to use with the file from the plurality of algorithms;

selecting an encryption key with a key value;

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file;

adding the file identifier to the file;

using the key value and the selected algorithm to encrypt the file and generate an encrypted file;

sending the encrypted file from a first person to a second person over the Internet in an e-mail message;

inputting a decryption key with a decryption key value;

validating the decryption key value with the key value associated with the file identifier; and

using the key value and the selected algorithm to decrypt the file.

70. The method as recited in claim 69, wherein the first person is the same as the second person.

71. A method of encrypting and decrypting a file with one of a plurality of algorithms, comprising the steps of:

selecting an algorithm to use with the file from the plurality of algorithms;

selecting an encryption key with a key value;

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file;

adding the file identifier to the file;

using the key value and the selected algorithm to encrypt the file and generate an encrypted file;

inputting a decryption key with a decryption key value;

validating the decryption key value with the key value associated with the file identifier; and

using the key value and the selected algorithm to decrypt the file;

wherein a portion of the file identifier is encrypted before it is inserted into the file.

72. The method as recited in claim 71, comprising the further step of decryption a portion of the file identifier before the decryption key value is validated.

73. The method as recited in claim 72, wherein all of the file identifier is encrypted before the decryption key value is validated.

75. A method of encrypting and decrypting a file with one of a plurality of algorithms, comprising the steps of:

selecting an algorithm to use with the file from the plurality of algorithms;

selecting an encryption key with a key value;

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file;

adding the file identifier to the file;

running a virus scan program on the file before it is encrypted;

using the key value and the selected algorithm to encrypt the file and generate an encrypted file;

inputting a decryption key with a decryption key value;

validating the decryption key value with the key value associated with the file identifier; and

using the key value and the algorithm to decrypt the file;

wherein a portion of the file identifier is encrypted before it is inserted into the file.

76. A method of encrypting a file with one of a plurality of algorithms, comprising the steps of:

selecting an algorithm to use with the file from the plurality of algorithms;

selecting an encryption key with a key value;

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file;

adding the file identifier to the file; and

uniquely identifying the encrypted file with an encrypted file header.

78. A method of encrypting and decrypting a file with one of a plurality of algorithms, the method comprising the steps of

selecting an algorithm to use with the file from the plurality of algorithms

selecting an encryption key with a key value

generating a file identifier from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file

adding the file identifier to the file

using the key value and the selected algorithm to encrypt the file and generate an encrypted file

sending the encrypted file from a first person to a second person in an e-mail message.

79. A method of encrypting and decrypting a file with one of a plurality of algorithms, the method comprising the steps of:

receiving an encrypted file from a first person by a second person in an e-mail message

extracting a file identifier from the file

inputting a decryption key with a decryption key value

validating the decryption key value with a key value associated with the file identifier

using the key value and the selected algorithm to decrypt the file.

SoCal IP Law Group

310 N. Westlake Blvd., Suite 120
Westlake Village, California 91362
phone +1 (805) 230-1350
fax +1 (805) 230-1355
info@socalip.com

Facsimile Cover Sheet and Certificate of Transmission Under 37 CFR 1.8

Appl. No.	:	09/259,991 Confirmation No. 5948
Appl. No. Bar Code	:	
Applicant	:	Mahne
Filed	:	03-01-1999
TC/A.U.	:	2134
Examiner	:	Matthew Smithers
Docket No.	:	M000-P02003US
Customer No.	:	33356
Customer No. Bar Code	:	
PTO Fax Number:	:	703/746-7238

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office on November 12, 2003 at 2:30PM PST.

By: _____
Joel G. Landau

List of Papers Enclosed:

1. Appeal Brief, 3 copies, 63 pages

Total Pages: 64